Amendments to the Claims:

This listing of claims will replace all prior versions, and listings, of claims in the application:

Listing of Claims:

1.-39. (Cancelled)

40. (Previously Presented) A computer implemented method for replicating a software application, the computer implemented method comprising:

executing the software application on a primary node to form a master application;

identifying resources and dependencies required by the master application to form required resources;

updating the required resources dynamically on the primary node;

generating a structure of the master application and a dynamic graph of the required resources from the required resources;

replicating the resources by transferring the structure to a set of secondary nodes via a network to form a replica; wherein the set of secondary nodes comprises one or more secondary nodes;

restoring the replica on the set of secondary nodes to form a set of clone software applications, wherein the set of clone software applications comprises one or more clone software applications;

executing the set of clone software applications on the set of secondary nodes, without loss of context; and

updating the set of clone software applications with incremental updates of the required resources of the master application to create a hot standby application.

41. (Previously Presented) The computer implemented method according to claim 40, wherein replicating the resources further comprises:

creating and maintaining a dependency tree, based on the dynamic graph, supplying, at all times, information on the replicated resources.

42. (Previously Presented) The computer implemented method according to claim 40, wherein replicating the resources further comprises:

checkpointing the resources on the set of secondary nodes, wherein the checkpointing having an adjustable period.

43. (Previously Presented) The computer implemented method according to claim 42, wherein replicating the resources further comprises:

capturing the resources on the primary node to create captured required resources;

transferring the captured required resources over the network to the set of secondary nodes; and

restoring the captured required resources on the set of secondary nodes.

44. (Previously Presented) The computer implemented method according to claim 42, wherein replicating the resources further comprises:

optimizing the checkpointing.

45. (Previously Presented) The computer implemented method according to claim 44, wherein the checkpointing is incremental.

46. (Previously Presented) The computer implemented method according to claim 44, wherein the checkpointing is discriminating.

47. (Previously Presented) The computer implemented method according to claim 42, wherein the checkpointing further comprises at least one of the following:

processing a synchronization barrier;

managing resources;

managing system resources; and

managing process resources.

48. (Previously Presented) The computer implemented method according to claim 40, wherein replicating the resources further comprises:

replicating applicative data files between the primary node, whereon the software application is run, and a stand-by node.

49. (Previously Presented) The computer implemented method according to claim 40, wherein replicating the resources further comprises:

ensuring functional continuity of the software application in a multi-computer architecture cluster, the software application being executed at a given time on one of the computers of the cluster, called the primary node, while other computers of the cluster are called a set of secondary nodes, wherein ensuring functional continuity further comprises:

replicating the software application on at least one of the secondary nodes to provide a set of clones of the application, wherein the set of clones comprises one or more clones;

updating the set of clones, and

responsive to detecting an event affecting the primary node, switching from the software application being executed on the primary node, to the software application being executed on the set of clones.

50. (Previously Presented) The computer implemented method according to claim 49, wherein replicating the software application is of a holistic nature.

51. (Previously Presented) The computer implemented method according to claim 49, wherein updating the set of clones further comprises updating the set of clones of the application.

52. (Previously Presented) The computer implemented method according to claim 49, wherein ensuring functional continuity further comprises supervising a state of the resources necessary to operate the software application.

53. (Previously Presented) The computer implemented method according to claim 49, wherein detecting an event affecting the primary node further comprises:

responsive to detecting an event affecting the primary node, electing a clone to be substituted for the primary node of the software application, wherein the secondary node on which the clone elect is installed becomes a new primary node.

54. (Previously Presented) The computer implemented method according to claim 53, wherein replicating the resources further comprises:

recording, on the set of clones, messages received by the primary node, the messages being injected into the clone elected as the new primary node when switching.

55. (Previously Presented) The computer implemented method according to claim 40, wherein replicating the resources further comprises:

optimization of information processing resources by load sharing and dynamic process distribution.

56. (Previously Presented) The computer implemented method according to claim 40, wherein replicating the resources further comprises:

performing non-interruptive maintenance by process relocation upon request, over a data-processing resource network.

57. (Previously Presented) The computer implemented method according to claim 40, wherein replicating the resources further comprises:

preserving applicative context in a mobile application.

58. (Currently Amended) A multi-computer system for ensuring functional continuity, capable of running, on at least one computer, at least one software application, the multi-computer system comprising:

a memory comprising a set of instructions;

a processor connected to the memory, capable of executing the set of instructions to implement a method comprising:

ensuring functional continuity of the software application in a multi-computer architecture cluster, the software application being executed at a given time on one of the computers of the cluster, called a primary node, to form a master application, while other computers of the cluster are called a set of secondary nodes, wherein ensuring functional continuity further comprises:

identifying resources and dependencies required by the master application to form required resources;

updating the required resources dynamically on the primary node;

generating a structure of the master application and a dynamic graph of the required resources from the required resources;

replicating the resources by transferring the structure to a set of secondary nodes via a network to form a replica; wherein the set of secondary nodes comprises one or more secondary nodes;

restoring the replica on the set of secondary nodes to form a set of clone software applications, wherein the set of clone software applications comprises one or more clone software applications;

executing the set of clone software applications on the set of secondary nodes, without loss of context; and

updating the set of clone software applications with incremental updates of the required resources of the master application to create a hot standby application.

~~replicating the software application on the set of secondary nodes to provide a set of clones of the application, wherein replicating the software application is of a holistic nature;~~

~~updating the set of clones, and responsive to detecting an event affecting the primary node, switching from the software application being executed on the primary node, to the software application being executed on the set of clones.~~